



PostgreSQL 12: What is in the kitchen?

Devrim Gündüz

Principal Systems Engineer @ EnterpriseDB

Founder @ GITC Pro Limited (UK) : <https://gitcpro.com>

Twitter : @DevrimGunduz



About me

- Who is this guy?
 - Using PostgreSQL since 1998.
 - London PostgreSQL Prime Minister
 - “The Guy With The PostgreSQL Tattoo”
 - Responsible for PostgreSQL YUM repository (<https://yum.postgresql.org>) where we host 200+ software
 - Major contributor to PostgreSQL
 - Fedora / CentOS packager for 50+ packages.
 - Father of 2 boys.
 - London, UK.

Twitter

Hashtag:

#PostgreSQL #PostgreSQLNL

Pease follow:

@PostgreSQLNL

@PostgreSQL

@PGConf_EU

Alternative Hashtag and account

#BlameMagnus
@BlameMagnus

Some of the new features

- Enabling JIT by default
- Pluggable storage
- DOS prevention for some commands
- Improved partitioning
- Integration of recovery.conf into postgresql.conf
- Various performance features

JIT

- Now enabled by default
- `auto_explain` and `EXPLAIN` exposes more info about JIT
- Red Hat / CentOS / Fedora requires `postgresql12-llvmjit`
- Debian / Ubuntu already bundles in core package.

Pluggable storage

- Just a start
- Usable parts will be in 13
- Options to use new storage engines

Integration of recovery.conf into postgresql.conf

- Finally...
 - PostgreSQL now throws an error if recovery.conf is found.
- (Almost) everything is moved into postgresql.conf
- recovery.signal
- standby.signal
- No more standby_mode
- recovery_target_timeline=latest by default

Integration of recovery.conf into postgresql.conf

- `trigger_file` → `promote_trigger_file`
- `pg_basebackup -R` appends to `postgresql.conf`
- A reload is enough for these parameters now:
 - `archive_cleanup_command`
 - `promote_trigger_file`
 - `recovery_end_command`
 - `recovery_min_apply_delay;`

psql improvements

- Add options for procedures in \df
- Show IP addresses in \conninfo
 - `$ psql -p 5412 -h localhost -U postgres -c "\conninfo"`

You are connected to database "postgres" as user "postgres" on host "localhost" (address "127.0.0.1") at port "5412".
 - `$ psql -p 5412 -U postgres -c "\conninfo"`

You are connected to database "postgres" as user "postgres" via socket in "/var/run/postgresql" at port "5412".
 - Useful if hostname resolves to multiple IP addresses

psql: URLs in \help

[postgres] # \help CREATE DATABASE

Command: CREATE DATABASE

Description: create a new database

Syntax:

CREATE DATABASE name

[[WITH] [OWNER [=] user_name]

[TEMPLATE [=] template]

<trimmed>

URL: <https://www.postgresql.org/docs/devel/sql-createdatabase.html>

psql tab complete improvements

- Lots of them
- ALTER DATABASE ... SET TABLESPACE
- Include partitioned tables in what's offered after ANALYZE.
- Include toast_tuple_target in what's offered after ALTER TABLE ... SET|RESET.
- Include HASH in what's offered after PARTITION BY.
- CREATE TABLE <name> with '(', OF or PARTITION OF.
- CREATE TABLE <name> OF with list of composite types.
- CREATE TABLE name (...) with PARTITION OF, WITH, TABLESPACE, ON

psql tab complete improvements

- Lots of them
- COMMIT (depending on the presence of a temporary table).
- - CREATE TABLE ON COMMIT with actions (only for temporary tables).
- SKIP_LOCKED option for VACUUM and ANALYZE
- ALTER INDEX ... ALTER COLUMN ... <column number goes here)
- Add completion for storage parameters after CREATE TABLE WITH
- Improve tab completion of ALTER INDEX/TABLE with SET STATISTICS in psql
- ...and a few more.

SKIP_LOCKED for VACUUM and ANALYZE

- Allows VACUUM to skip the work on a relation if there is a conflicting lock on it when trying to open it at the beginning of its processing.
- Note: v11: Vacuum can process multiple tables
- vacuumdb also now has --skip-locked.

SKIP_LOCKED for VACUUM and ANALYZE

- `[pagila] # VACUUM (FULL, SKIP_LOCKED,VERBOSE)`
`film,language;`

WARNING: skipping vacuum of "film" --- lock not available

INFO: vacuuming "public.language"

INFO: "language": found 0 removable, 6 nonremovable row versions in 1 pages

DETAIL: 0 dead row versions cannot be removed yet.

CPU: user: 0.00 s, system: 0.00 s, elapsed: 0.07 s.

VACUUM

vacuumdb: --min-xid-age and --min-mxid-age

- Improves the selectivity of the relations to vacuum or analyze
- Transaction ID or multixact ID
- Chance to prioritize tables to prevent wraparound of one or the other.
- When used with `-table`, opportunity to target subset of tables

max_connections and max_wal_senders

- max_wal_senders is no more a part of max_connections
- No more blocking of base backups, if there are enough max_wal_senders are available
- Now it is like autovacuum and bgworkers.

pg_dumpall and pg_dump

- `pg_dumpall`: Now can exclude a database from `pg_dumpall`
 - Can be given once
 - The argument can be a pattern including wildcard characters.
- `pg_dump`: Include `ALTER INDEX ... ALTER COLUMN ... SET STATISTICS` info
-
- `pg_dump`: Now allows multiple rows per insert
 - Useful to speed up loading data in a different database engine.
 - `pg_dump dbname --inserts --rows-per-insert=250`
(Default:100)

pg_hba.conf

- New clientcert hba option: verify-full
- This allows a login to require both that the CN of the certificate matches (like authentication type cert) *and* that another authentication method (such as password or kerberos) succeeds as well.

The old value of clientcert=1 maps to the new clientcert=verify-ca,

- clientcert=0 maps to the new clientcert=no-verify
- verify-full will add the validation of the CN.

Some more features

- Hyperbolic funcs, and log(10) support
 - The SQL:2016 standard : sinh(), cosh(), and tanh().
 - log(10) is also exposed.
- Scan GiST indexes in physical order during VACUUM
 - Faster scan of the index (as compared to logical walking it in logical order), because seq I/O is faster.
- Set fallback_application_name for a walreceiver to cluster_name
 - Default is walreceiver, makes it harder to distinguish when there are multiple replicas
Use cluster_name instead, if it is set.
 - Using primary_conninfo still works.

postgresql.conf changes and updates

- New parameters:
 - `ssl_min_protocol_version = 'TLSv1'`
 - `ssl_max_protocol_version = ''`
 - `shared_memory_type` (Alternatives: `nmap`, `sysv`, `windows`)
 - `log_statement_sample_rate = 1`
 - `plan_cache_mode = auto` (Alternatives: `force_generic_plan`, `force_custom_plan`)
 - Executor level, not planner level
- Updated parameters:
 - `autovacuum_vacuum_cost_delay`: 200 ms → 2 ms
 - `extra_float_digits` : 0 → 1
 - `jit = off` → `on`

Some more features

- Avoid some rewrites in “ALTER TABLE ... SET DATA TYPE timestamp”
 - When the timezone is UTC, timestamptz and timestamp are binary coercible, avoid the table rewrite, and continue to needlessly rewrite any index on an affected column.
- Log PostgreSQL version number on startup
 - 2019-03-14 11:33:39.842 CET [7829] LOG: starting PostgreSQL 12devel on x86_64-pc-linux-gnu, compiled by gcc (GCC) 8.3.1 20190223 (Red Hat 8.3.1-2), 64-bit
 - 2019-03-14 11:33:39.842 CET [7829] LOG: listening on IPv4 address "127.0.0.1", port 5412
 - 2019-03-14 11:33:39.916 CET [7829] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5412"
 - 2019-03-14 11:33:40.059 CET [7829] LOG: listening on Unix socket "/tmp/.s.PGSQL.5412"

Some more features

- Add more columns to `pg_stat_ssl`
 - `client_serial` and `issuer_dn`
 - Allows uniquely identify a client cert.
 - Rename `clientdn` to `client_dn` for consistency.
- Scan GiST indexes in physical order during VACUUM
 - Faster scan of the index (as compared to logical walking it in logical order), because seq I/O is faster.

Some more features

- Allow COPY FROM to filter data using WHERE conditions
 - Extends the COPY FROM command with a WHERE condition, which allows doing various types of filtering while importing the data (random sampling, condition on a data column, etc.).
 - Low overhead
- pg_upgrade: --socketdir option
 - This allows control of the directory in which the postmaster sockets are created for the temporary postmasters started by pg_upgrade
 - Useful for long path names.

Some more features

- Do not log empty incomplete startup packet
- Change "checkpoint starting" message to use "wal"
- Add support for nearest-neighbor (KNN) searches to SP-GiST
 - Previously only GiST.
 - Provided for three SP-GiST opclasses: `quad_point_ops`, `kd_point_ops` and `poly_ops`

pg_stat_statements updates

- `pg_stat_statements_reset` can now reset statistics specific to a particular user/db/query.
- Now, it can discard the statistics gathered so far by `pg_stat_statements` corresponding to the specified `userid`, `dbid`, and `queryid`.
- If no parameter is specified or all the specified parameters have default value 0, it will discard all statistics as per the old behavior.

What is old in PostgreSQL 12?

- abstime, reltime, tinterval datatypes are removed (already deprecated)
- timetravel extension is removed (Use PL/pgSQL)
 - (Used to keep the insert and delete times for each tuple)

What is in the kitchen?

- Undo logs (will not make into v12)
- Online verification of checksums
- Progress report of pg_checksums
- Offline enabling/disabling of checksums
- Some indexing improvements by Peter G.
- psql: Display foreign keys
- Avoid creation of the free space map for small tables
- Monitoring progress of CREATE INDEX [CONCURRENTLY]
- Query logging of prepared statements
- New vacuum option to do only freezing
- Online change of primary_conninfo



PostgreSQL 12: What is in the kitchen?

Devrim Gündüz

Principal Systems Engineer @ EnterpriseDB

Founder @ GITC Pro Limited (UK)

Twitter : @DevrimGunduz

